Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Отчёт по лабораторной работе №1 по теме:

**«Матричные способы представления графов»**

по дисциплине

«Дискретная математика»

Выполнили:

ст. группы 245

Бригады №6

Комарова Ксения

Евдокимов Александр

Проверила:

Доц. Бакулева Марина Алексеевна

Рязань 2024

# Цель работы

Целью работы является изучение матричных способов представления

графов.

**Практическая часть**

1. Мы составили алгоритм программы, реализующей перевод из матрицы инцидентности (7 на 14) графа в матрицу смежности и обратно, учитывая при этом исходный тип графа (неориентированный, ориентированный, смешанный). Результат представлен на рисунках 1-3.

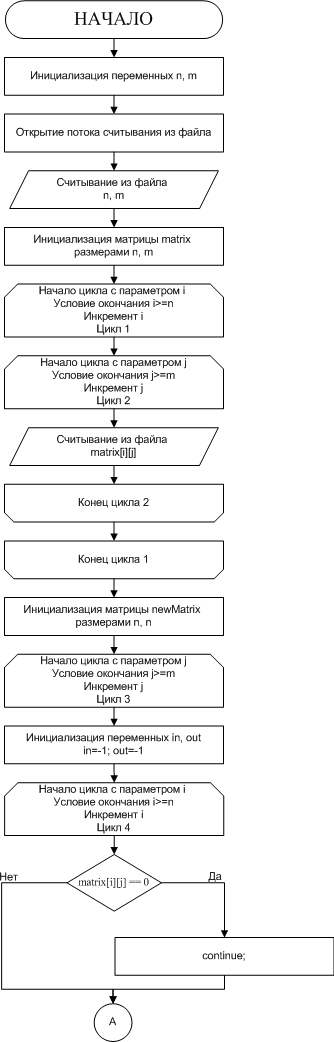


Рисунок 1 – Алгоритм программы(1)

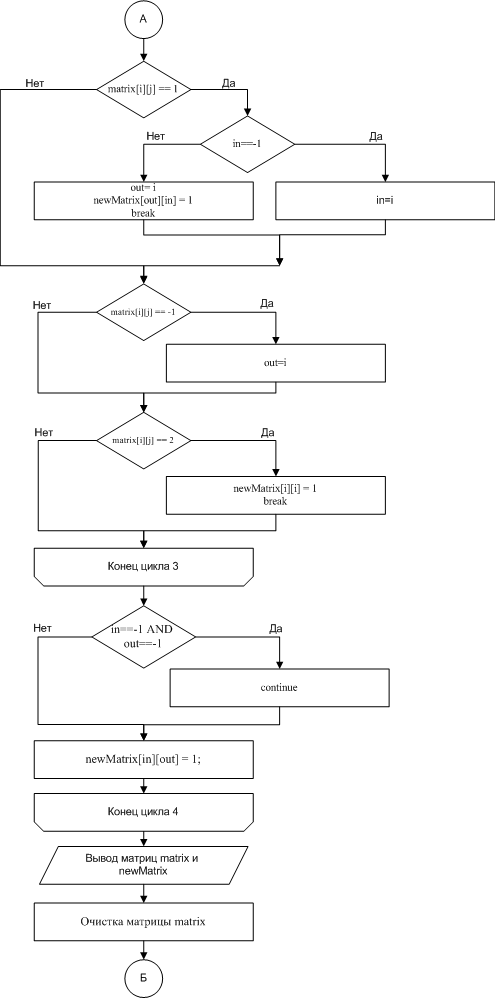


Рисунок 2 – Алгоритм программы(2)

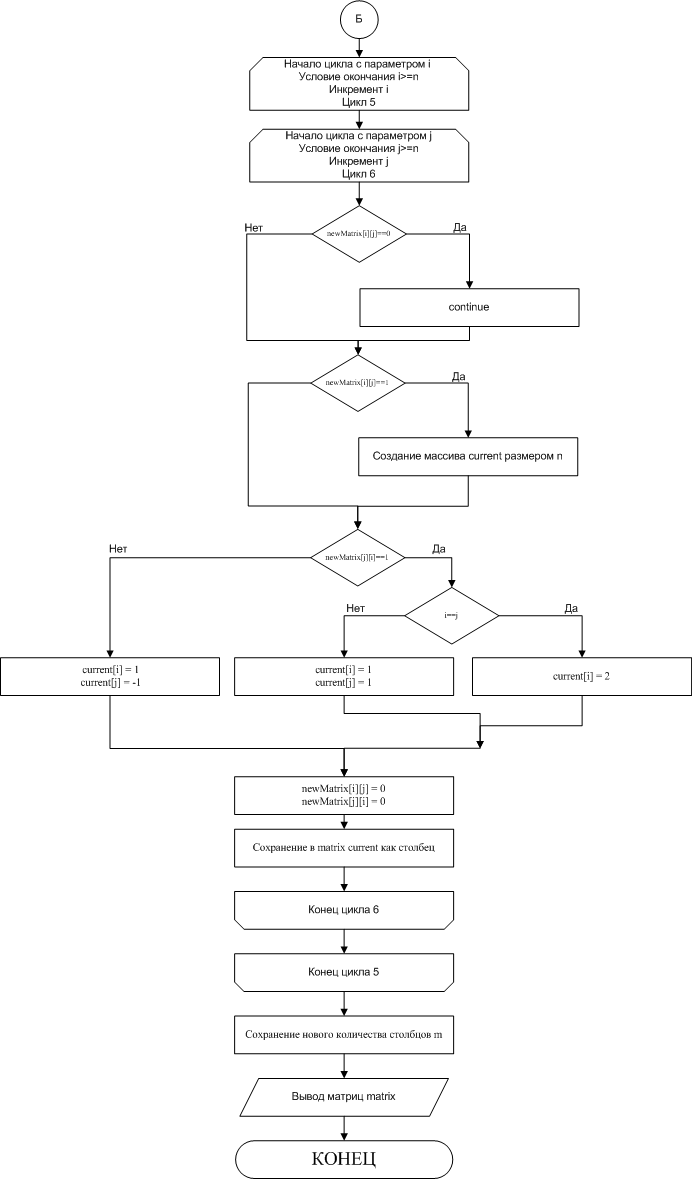


Рисунок 3 – Алгоритм программы(3)

1. Была создана программа на языке C++, реализующая перевод из заданного способа матричного представления графа в другой (приложение А).
2. Контрольный пример (таблицы 1-2) и результаты машинного расчёта приведены ниже (рисунок 2).

Таблица 1 - Матрица идентичности

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | а1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | a14 |
| x1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 1 |
| x2 | -1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x3 | 0 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | -1 |
| x4 | 0 | 0 | 0 | 1 | +-1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x5 | 0 | -1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | -1 | 1 | 0 | 0 |
| x6 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | -1 | 0 | 0 | 1 | 0 |
| x7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

Таблица 2 - Матрица смежности при ручном расчёте

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | x1 | x2 | x3 | x4 | x5 | x6 | x7 |
| x1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| x2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| x3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| x4 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| x5 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| x6 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| x7 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

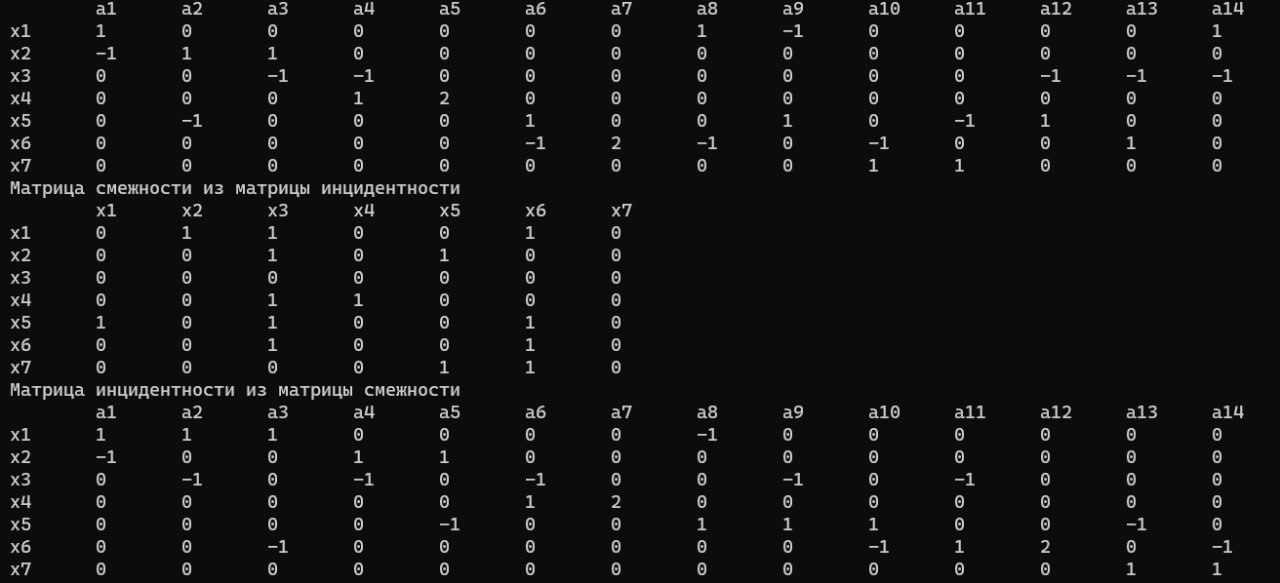


Рисунок 2 – Результат выполнения программы

**Заключение**

В данной лабораторной работе мы изучили матричные способы представления графов.

Приложение А.Код программы

#include <iostream>

#include <vector>

#include <fstream>

typedef std::vector<short int> iVect;

void PrintMtr(std::vector <iVect>& matrix, char nameVert, char nameHor, int n, int m) {

std::cout << "\t";

for (int i = 0; i < m; i++) {

std::cout << nameHor << i + 1 << "\t";

}

std::cout << "\n";

for (int i = 0; i < n; i++) {

std::cout << nameVert << i + 1 << "\t";

for (int j = 0; j < m; j++)

std::cout << matrix[i][j] << "\t";

std::cout << "\n";

}

}

int main()

{

setlocale(LC\_ALL, "");

int n, m;

std::ifstream input("Data/input.txt");

if (!input) {

std::cout << "Файл не найден\n";

return 0;

}

input >> n >> m;

std::vector<iVect> matrix(n, iVect(m, 0));

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

input >> matrix[i][j];

std::vector<iVect> newMatrix(n, iVect(n, 0));

for (int j = 0; j < m; j++) {

int in=-1, out=-1;

for (int i = 0; i < n; i++) {

if (matrix[i][j] == 0) continue;

if (matrix[i][j] == 1) {

if (in == -1)

in = i;

else { out = i; newMatrix[out][in] = 1; break; }

}

if (matrix[i][j] == -1) out = i;

if (matrix[i][j] == 2) { newMatrix[i][i] = 1; break; }

}

if (in==-1 && out==-1) continue;

newMatrix[in][out] = 1;

}

std::cout << "Матрица инцидентности\n";

PrintMtr(matrix, 'x', 'a', n, m);

std::cout << "Матрица смежности из матрицы инцидентности\n";

PrintMtr(newMatrix, 'x', 'x', n, n);

matrix.clear();

for(int i = 0; i<n; i++)

for (int j = 0; j < n; j++) {

if (newMatrix[i][j] == 0) continue;

if (newMatrix[i][j] == 1) {

iVect current(n, 0);

if (newMatrix[j][i] == 1) {

if(i==j)

current[i] = 2;

else {

current[i] = 1;

current[j] = 1;

}

}

else if (newMatrix[j][i] == 0) {

current[i] = 1;

current[j] = -1;

}

newMatrix[i][j] = 0;

newMatrix[j][i] = 0;

matrix.push\_back(current);

}

}

m = matrix.size();

std::cout << "Матрица инцидентности из матрицы смежности\n\t";

for (int i = 0; i < m; i++) {

std::cout << 'a' << i + 1 << "\t";

}

std::cout << "\n";

for (int i = 0; i < n; i++) {

std::cout << 'x' << i + 1 << "\t";

for (int j = 0; j < m; j++)

std::cout << matrix[j][i] << "\t";

std::cout << "\n";

}

return 0;

}